

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Miroslav Matijević

Miselna igra za platformo Google Cardboard

DIPLOMSKO DELO

VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM PRVE
STOPNJE RAČUNALNIŠTVO IN INFORMATIKA

Ljubljana, 2017

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Miroslav Matijević

Miselna igra za platformo Google Cardboard

DIPLOMSKO DELO

VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM PRVE
STOPNJE RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: viš. pred. dr. Aljaž Zrnec

Ljubljana, 2017

Rezultati diplomskega dela so intelektualna lastnina avtorja in Fakultete za računalništvo in informatiko Univerze v Ljubljani. Za objavljanje ali izkoriščanje rezultatov diplomskega dela je potrebno pisno soglasje avtorja, Fakultete za računalništvo in informatiko ter mentorja.

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

Uporabnost mobilnih telefonov že dolgo ni omejena zgolj na telefoniranje in pošiljanje SMS spočil. Že kar nekaj časa poznamo vrsto pametnih aplikacij, ki nas spremljajo skozi vsakdanje življenje, a nedolgo nazaj smo stopili korak naprej in spoznali še bolj realistično predstavitev podatkov s pomočjo VR (virtual reality) očal. V okviru diplomske naloge predstavite tehnologije in funkcionalnosti, ki jih ponuja platforma Google Cardboard, nato pa predstavite primer uporabe platforme z izdelavo miselne igre. Predstavite tudi orodja in okolja, ki ste jih v okviru praktičnega dela uporabljali.

Izjava o avtorstvu diplomskega dela

Spodaj podpisani Miroslav Matijević, z vpisno številko 63110065, sem avtor diplomskega dela z naslovom:

Miselna igra za platformo Google Cardboard

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom viš. pred. dr. Aljaž Zrneca,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela,
- soglašam z javno objavo elektronske oblike diplomskega dela na svetovnem spletu preko univerzitetnega spletnega arhiva.

V Ljubljani, dne 11. februarja 2017

Podpis avtorja:

Zahvaljujem se mentorju viš. pred. dr. Aljažu Zrncu za strokovno pomoč pri izdelavi diplomskega dela. Prav tako se zahvaljujem svoji družini, puncu in prijateljem, ki so me spodbujali in podpirali v času izdelave diplomskega dela. Zahvaljujem se tudi vsem tistim, ki so preizkusili mojo aplikacijo in mi podali svoje mnenje in koristne nasvete.

Kazalo

Povzetek

Abstract

Poglavje 1	Uvod	1
Poglavje 2	Navidezna resničnost.....	3
2.1	Kaj pomeni navidezna resničnost	3
2.2	Kratek pregled zgodovine	3
2.3	Najbolj znani sistemi za VR	6
2.3.1	Oculus Rift.....	6
2.3.2	HTC Vive	7
2.3.3	Samsung gear VR	7
2.3.4	Google Cardboard	8
Poglavje 3	Uporabljene naprave in programska oprema	9
3.1	Naprave in komponente	9
3.1.1	Mobilna naprava	9
3.1.2	Žiroskop.....	10
3.1.3	VR naprava	10
3.2	Programska oprema	11
3.2.1	Unity	11
3.2.2	C#	12
3.2.3	SQLite.....	12
3.2.4	DB Browser for SQLite	13
3.2.5	PhotoFiltre	13
Poglavje 4	Razvoj aplikacije	15

4.1	Ideja	15
4.2	Načrtovanje	15
4.3	Grafično oblikovanje	17
4.3.1	3D oblikovanje	17
4.3.2	2D oblikovanje	18
4.4	Zvok	19
4.5	Kodiranje	20
4.6	Podatkovna baza	22
4.6.1	Opis podatkovne baze	22
4.6.2	Branje iz podatkovne baze	24
4.6.3	Pisanje v podatkovno bazo	24
Poglavje 5	Delovanje aplikacije	27
5.1	Osnovne lastnosti aplikacije	27
5.2	Zagon aplikacije	27
5.3	Uporabniški vmesnik	29
5.3.1	Vnos imena	29
5.3.2	Glavni meni	31
5.3.3	Rezultati	31
5.3.4	Navodila	32
5.3.5	Animacija	33
5.4	Igralni del aplikacije	34
5.4.1	Igra	34
5.4.2	Izid igre	35
Poglavje 6	Sklepne ugotovitve	37
Literatura	39

Seznam uporabljenih kratic

Kratica	angleško	Slovensko
2D	two dimensional	dvo dimenzionalni
3D	three dimensional	tri dimenzionalni
CLI	common language infrastructure	skupni jezik infrastrukture
ECMA	european computer manufacturers association	evropska skupnost računalniških proizvajalcev
GC	Google Cardboard	Googlova lepenka
GPS	global positioning system	globalni sistem pozicioniranje
GVR	Google virtual reality	Googlova navidezna resničnost
ISO	international organization for standardization	mednarodna organizacija za standardizacijo
OS	operating sistem	operacijski sistem
SDK	software development kit	razvojno programsko okolje
USB	universal serial bus	univerzalno serijsko vodilo
VR	virtual reality	navidezna resničnost

Povzetek

Naslov: Miselna igra za platformo Google Cardboard

Mobilne naprave, kot so pametni mobilni telefoni, se že dolgo ne uporabljajo zgolj za klicanje in sporočanje. Zadnja leta opazamo velik premik na področju podpore navidezne resničnosti na pametnih telefonih. Vse več je dodatkov in aplikacij, ki pametni telefon spremenijo v napravo, ki omogoča vstop v svet navidezne resničnosti. V diplomskem delu so predstavljene tehnologije za podporo navidezne resničnosti, ter opisan proces razvoja aplikacije za mobilno platformo Android. Aplikacija omogoča igranje igre v navidezni resničnosti, z uporabo Google očal za navidezno resničnost in pametnega mobilnega telefona. V diplomskem delu so prav tako opisana tudi programska orodja, postopki in prijemi, ki so uporabljeni za razvoj aplikacije.

Ključne besede: navidezna resničnost, Google lepenka, Android, igra, očala, mobilna naprava.

Abstract

Title: Puzzle game for the Google Cardboard platform

Mobile devices, like mobile smartphones, are no longer used only for calling and messaging. Last year's we see great progress in the field of support for virtual reality on smartphones. More and more add-ons and applications are transforming smartphones into device, which allows entry into the world of virtual reality. In this graduate thesis are presented the technologies to support virtual reality, and describes the process of developing applications for the Android mobile platform. The application allows user to play a game in virtual reality, using Google glasses for virtual reality and smart mobile phone. In the following thesis are also described the software tools, methods and approaches that are used to develop applications.

Keywords: virtual reality, Google Cardboard, Android, game, glasses, mobile device

Poglavje 1 Uvod

Igre so prisotne v vseh obdobjih našega življenja. Že kot otroci smo se igrali v peskovniku, kasneje na igriščih in v telovadnicah. Že nekaj desetletij pa so vsem znane igre na osebnih računalnikih, ki so se kasneje preselile tudi na mobilne naprave. Cilj velikih proizvajalcev v industriji iger je narediti igro, ki bo čim bolj zanimiva in privlačna za igralca. Zelo dober način, za doseganje tega učinka je razvoj igre, ki je čim bolj resnična. Proizvajalci iger nenehno izboljšujejo grafične pogone iger ter s tem ustvarjajo bolj resnične učinke v igrah.

Tehnologija navidezne resničnosti, kot jo poznamo v današnjem času, je povečala privlačnost iger. Proizvajalcem iger ponuja dodaten način izdelave bolj resnične igre, uporabnikom pa boljšo igralno izkušnjo s sprejemljivimi stroški opreme. Resničnost so dvignili na novo raven, ki se imenuje navidezna resničnost (ang. Virtual Reality). Kot naprava, ki nam omogoča vstop v navidezno resničnost, se danes najpogosteje pojmujejo očala za navidezno resničnost.

V okviru diplomskega dela je opisan razvoj miselne igre za mobilne naprave z operacijskim sistemom Android. Igra je izdelana z uporabo tehnologiji za navidezno resničnost. Za igranje te igre mora uporabnik imeti Google Cardboard očala za navidezno resničnost. Za ta očala smo se odločili, ker so cenovno najbolj ugodna, lahko pa jih izdelamo tudi kar doma. Za igranje poleg očal potrebujemo še dovolj zmogljiv pametni telefon z operacijskim sistemom Android.

Glavni cilj diplomskega dela je izdelati čim bolj enostavno igro, ki za igranje, poleg očal, ne potrebuje nobenih dodatnih naprav. Gre za igranje, zgolj z uporabo v mobilno napravo vgrajenega senzorja žiroskop, ki zaznava vsak premik očal oz. glave in na podlagi teh premikov izvaja aktivnosti v igri. Tak način igranja za razvijalca predstavlja dodaten izziv, saj je nadzor nad dogajanjem v igri čisto drugačen od tistega, ki smo ga navajeni pri računalniških igrah, kjer imamo na voljo tipkovnico in miško.

Glede na to, da je tehnologija navidezne resničnosti še dokaj mlada veja v mobilni industriji, je v trgovini z mobilnimi aplikacijami teh aplikacij že ogromno. Velik del vseh aplikacij za navidezno resničnost predstavljajo igre, saj je na tem področju navidezna resničnost še posebej uporabna. Ob pregledu aplikacij v spletni trgovini je zaslediti veliko iger, ki imajo dobro razvit osrednji del igre, kateri se tudi največ uporablja. Ostali deli aplikacije kot so meniji, navodila, rezultati, nastavitve aplikacije pa so slabše razviti. Zaradi ugotovitev med pregledom drugih

aplikacij je bil sekundarni cilj diplomskega dela izdelati celostno igro, brez pomanjkljivosti, ki smo jih zasledili pri drugih igrah. Torej izdelati igro, ki bo imela poleg glavnega – igralnega dela, tudi ostale dele igre razvite v tehnologiji navidezne resničnosti. Aplikacija naj bi tudi uporabljala podatkovno bazo za beleženje rezultatov in shranjevanje odigranih iger.

Poglavje 2 Navidezna resničnost

2.1 Kaj pomeni navidezna resničnost

Navidezna resničnost je znana tudi kot računalniško ustvarjena resničnost. Tehnologija oponaša resnični svet in uporabnika v njem. Omogoča uporabo preko dveh ali enega zaslona, ki je razdeljen na dva dela za vsako oko eden [1]. Tehnologije in naprave, ki nam omogočajo vstop v navidezno resničnost se med seboj razlikujejo po načinu generiranja navidezne resničnosti. Večina teh naprav potrebuje za delovanje računalnik ali mobilno napravo, obstajajo pa tudi take, ki imajo računalnik že vgrajen.

2.2 Kratek pregled zgodovine

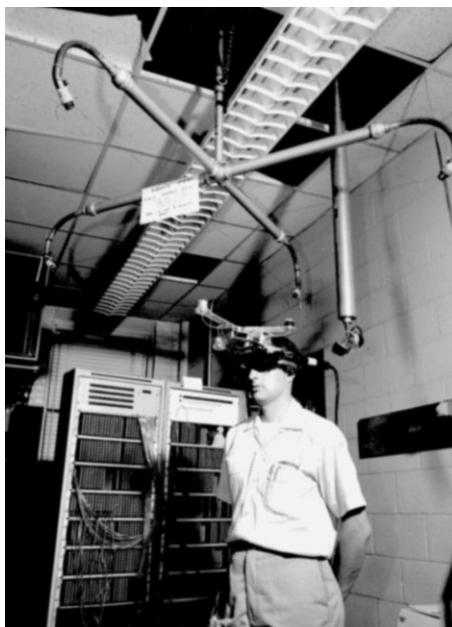
Pojem navidezna resničnost je znan že od leta 1935. Takrat je Stanley G. Weinbaum v kratki zgodbi "Pygmalion's Spectacles" opisal VR (virtual reality) sistem, ki temelji na očalih za VR, ki so omogočala s pomočjo holografskih zapisov doživljanje resničnih dogodkov z dotikom in vonjem [2].

Morton Leonard Heilig je prvo napravo za VR patentiral leta 1950. Leta 1962 je naredil prototip, ki ga je poimenoval Sensorama. Ta naprava je uporabniku omogočala predvajanje petih kratkih filmov. Končni izdelek je razen predvajanja slike in zvoka, omogočal tudi vklop vonja in dotika. Sensorama je popolnoma mehanični stroj, ki je ohranjen v delujočem stanju do današnjega časa. Prototip prikazuje Slika 2.1.



Slika 2.1: Sensorama

Ivan Sutherland in njegov študent Bob Sproulli, sta leta 1968 naredila prvi stroj za obogateno resničnost (ang. Augmented reality), ki je uporabljal precej podoben princip delovanja, kot ga uporabljajo današnje VR naprave. Stroj je bil precej okoren, saj je visel s stropa na vodilu in na koncu le-tega so bila očala. Stroj je za prikaz uporabljal zelo enostavne geometriske oblike, ki so bile izdelane s pomočjo vektorjev (vire frame rooms). Zaradi opazne velikosti in položaja pri uporabi, so ga poimenovali Damokov meč (The Sword of Damocles) [3]. Stroj prikazuje Slika 2.2.



Slika 2.2: Damokov meč

Za omenjenim izumom je bilo nekaj zatišja na tem področju, in sicer do leta 1990, ko je Jonathan Waldern predstavil VR sistem Virtuality za igranje arkadne igre. Leta 1991 je podjetje Sega razvilo sistem za navideznoresničnost SEGA VR, ki pa ni dosegel prodajnih polic, ker je uporabnikom povzročal glavobole. Uspelo jim je leta 1994, ko so na podlagi prejšnjega prototipa izdelali model Sega VR-1, ki se je prodajal v sklopu njihovega arkadnega stroja sega model 1. Tudi podjetje Nintendo je leta 1995 začelo prodajati konzolo za navidezno resničnost. Istega leta so se začeli prodajati tudi VR sistemi VFX1, ki so bili namenjeni osebnim računalnikom [4]. VR sistem VFX1 prikazuje Slika 2.3.



Slika 2.3: VR sistem VFX1

Podjetnik Palmer Luckey je leta 2010 ustanovil podjetje Oculus VR in naredil prvi prototip sistema Oculus Rift. Leta 2013 je izšla Vendetta Online, prva igra za Oculus Rift [5]. Leta 2014 je svetovno znano podjetje Facebook, kupilo celotno podjetje Oculus VR za 2 milijardi dolarjev [6]. Istega leta je podjetje Google izdalo svojo cenovno ugodnejšo različico VR očal, ki delujejo v kombinaciji s pametnim mobilnim telefonom. Podjetji HTC in Valve sta zaradi potreb trga sklenili sodelovanje in leta 2016 začeli s prodajo VR sistema HTC Vive.

2.3 Najbolj znani sistemi za VR

V današnjem času obstaja veliko sistemov, ki nam omogočajo vstop v navidezno resničnost. V sledečem delu diplomskega dela, je opisanih le nekaj najbolj znanih sistemov za VR, ki se uporabljajo bodisi za računalnike ali za mobilne naprave.

2.3.1 Oculus Rift

Podjetje Oculus VR je nekje v sredini leta 2013 izdelalo prvi prototip VR sistema, ki so ga poimenovali Development Kit 1. Približno leto kasneje so izdelali drugi prototip, ki se je imenoval Development Kit 2. Končna različica VR sistema Oculus Rift, je izšla leta 2016 in leta omogoča priklop na računalnik preko USB (universal serial bus) priključka. VR sistem vsebuje senzorje žiroskop, magnetometer in merilec pospeškov za zaznavanje orientacije in obračanja očal. Poleg teh senzorjev sistem uporablja sistem za pozicioniranje in rotacijski sistem. Rotacijski sistem zaznava obračanje uporabnika v prostoru. Sistem za pozicioniranje pa zaznava položaj očal v prostoru. Ta dva sistema s pomočjo IR žarkov, ki jih oddajajo očala, omogočata spremljanje položaja uporabnika. VR sistem uporablja tudi ročne kontrole, ki jih sestavljata dve napravi (za vsako roko ena). Na vsaki napravi se nahajata dve tipki in igralna palčica. VR sistem ima slušalke, preko katerih je možno poslušati 3D (three dimensional) zvok [7]. Cena tega produkta se giblje okoli 600 \$. Sistem Oculus Rift prikazuje Slika 2.4.



Slika 2.4: Oculus Rift

2.3.2 HTC Vive

Sistem HTC Vive za VR je izdelan v sodelovanju podjetij HTC in Valve. Prvi prototip VR sistema HTC Vive je bil izdelan v letu 2014. Prodaja produkta se je začela konec marca leta 2016. Priklop in delovanje tega VR sistema je podobno Oculus VR sistemu, le da je produkt podjetij HTC in Valve mnogo bolj dovršen. Na njem se nahaja približno 70 senzorjev, ki zaznavajo pozicijo uporabnika. Uporablja tudi dva ločena zaslona za prikaz 3D slike. Varnost uporabnika je zagotovljena s sistemom, ki zaznava in opozarja na predmete v okolici in s tem preprečuje poškodbe uporabnika. Prav tako kot VR sistem Oculus Rift, tudi HTC Vive vsebuje dve ročni kontroli. VR sistem HTC Vive je v celoti podprt s strani podjetja Valve, ki zanj izdeluje računalniške igre. Cena tega VR sistema se giblje okoli 800 \$. Sistem HTC Vive prikazuje Slika 2.5.



Slika 2.5: HTC Vive

2.3.3 Samsung gear VR

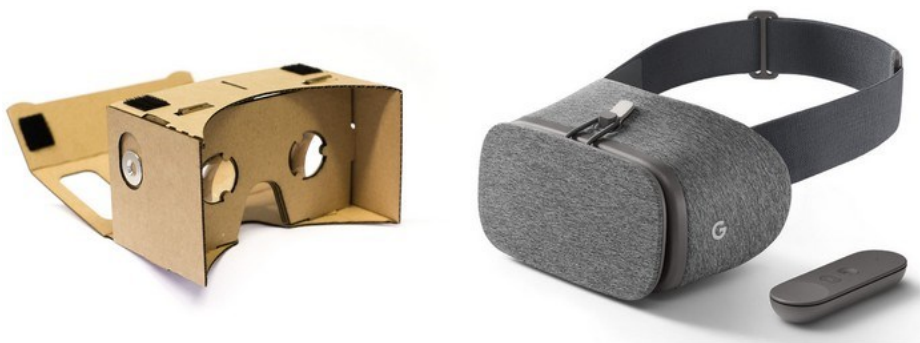
Sistem za VR Samsung gear VR, je produkt podjetja Samsung v sodelovanju s podjetjem Oculus. VR sistem se je uradno začel prodajati konec leta 2016. VR sistem podpira mobilne naprave višjega cenovnega razreda, kot so Samsung Galaxy S6 in novejši modeli iz družine S. VR sistem deluje tako, da nanj pritrdimo mobilno napravo. Ta vsebuje senzorje, ki sporočajo pozicijo v prostoru. VR sistem ima tudi nekaj gumbov pri strani, ki uporabniku omogočajo dodatni nadzor. Cena takšnega VR sistema se giblje okoli 100 \$ [8]. Sistem Samsung gear VR prikazuje Slika 2.6.



Slika 2.6: Samsung gear VR

2.3.4 Google Cardboard

Cenovno najugodnejši VR sistem za mobilne naprave, je bil izdelan leta 2014 s strani podjetja Google. Sistem za VR je sestavljen iz kartona. Ima še leče, ki so lahko različnih velikosti. Edina dodatna kontrola je magnet pri strani kartonskih očal, ki ob premiku deluje na kompas v mobilni napravi. VR sistem podpira mobilne naprave do velikosti 6 inčev. Na spletu je veliko nasvetov za izdelavo očal, saj si uporabnik lahko tak VR sistem izdelava kar doma. Za zaznavanje pozicije se uporablja senzor žiroskop, ki je vgrajen v nekaterih mobilnih napravah. V primeru, da mobilna naprava nima žiroskopa, nam večina aplikacij na njej ne deluje. Cardboard SDK (Software development kit) podpirata Android in iOS mobilna operacijska sistema. Google je v začetku leta 2016, posodobil VR platformo Google Cardboard z izdajo izpopolnjene različice, ki se imenuje Daydream [9]. Sistema Cardboard in Daydream prikazuje Slika 2.7.



Slika 2.7: Cardboard in Daydream

Poglavje 3 Uporabljene naprave in programska oprema

3.1 Naprave in komponente

V tem poglavju so opisane naprave, ki smo jih uporabili v okviru razvoja aplikacije. Prav tako so predstavljene komponente naprav, ki so pomembne za razvoj aplikacije za platformo GVR (Google virtual reality).

3.1.1 Mobilna naprava

V sklopu diplomskega dela je bila uporabljena mobilna naprava Samsung Galaxy S6 SM-G920F. Na napravi je nameščen OS (operating sistem) Android različice 6.0.1 (Marshmallow). Naprava uporablja zaslon velikosti 5.1 inča, kar je še v okviru zahtev VR naprave, ki je bila uporabljena. Mobilna naprava podpira senzor žiroskop (ang. Gyro sensor), ki je ključnega pomena za razvijanje aplikacije za mobilne VR naprave. Na omenjeni mobilni napravi so bili izvedeni vsi testi v okviru razvojnega procesa. Mobilno napravo Samsung Galaxy S6 prikazuje Slika 3.1.



Slika 3.1: Mobilna naprava Samsung Galaxy S6

3.1.2 Žiroskop

Žiroskop je senzor, ki ga pametni telefoni uporabljajo za ohranjanje in nadzor položaja. Deluje na principu vrtilne količine. Določa usmeritev naprave, v primeru vrtenja ali nagibanja. V mobilnih napravah je ključen predvsem pri aplikacijah, za katere je pomembna usmeritev naprave. V skupino takšnih aplikacij spadajo aplikacije za navigacijo in skoraj vse 3D aplikacije. Po navadi deluje v sodelovanju s senzorjem pospeška in GPS (Global Positioning System) senzorjem [10]. Žiroskop senzor prikazuje Slika 3.2.



Slika 3.2: Žiroskop senzor

3.1.3 VR naprava

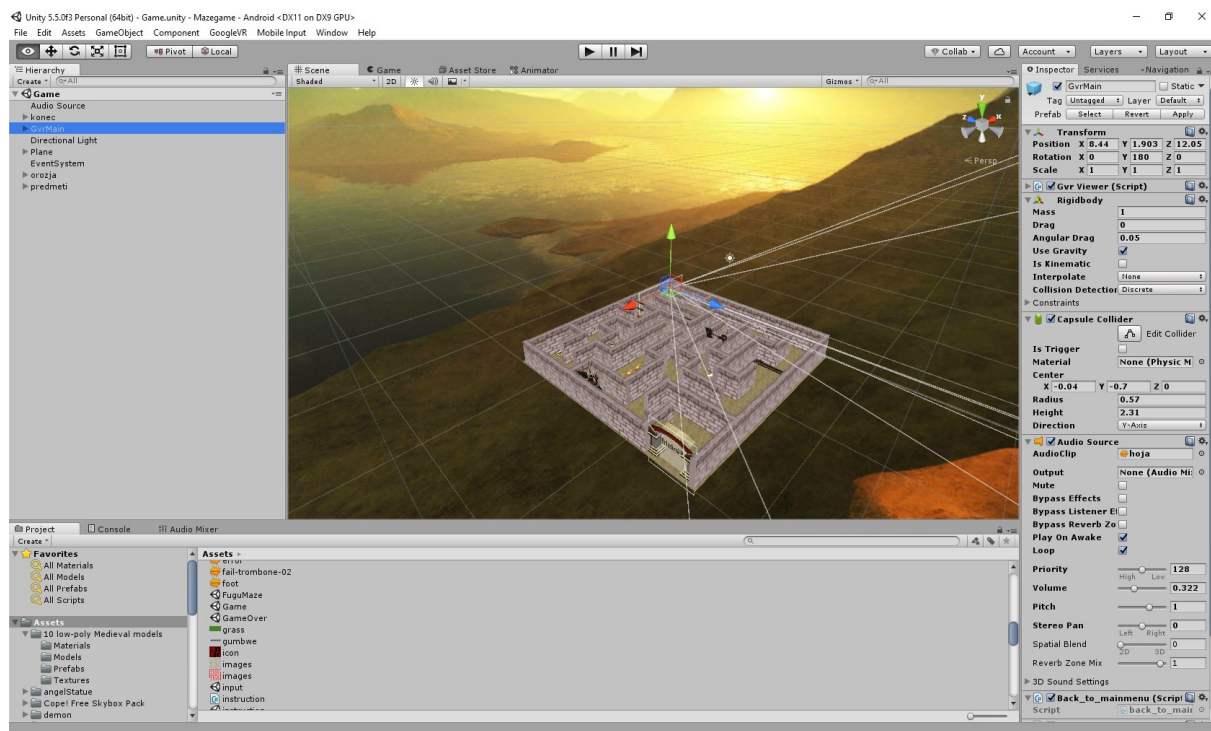
Za razvoj diplomskega dela so bila uporabljena VR očala Google Cardboard. Verzija uporabljenih VR očal je 2.0. Ta verzija GC (Google Cardboard) očal za VR, uporabniku ponuja le eno dodatno fizično kontrolo. Uporablja se jo predvsem za potrditve v aplikaciji, s posrednim dotikom zaslona mobilne naprave, ki ga izvedemo preko aluminijaste folije nameščene na robu VR očal. Omenjena funkcija VR naprave ni bila izkoriščena za potrebe diplomskega dela, ker je bila ideja izdelati aplikacijo, ki bo delovala na vseh GC očalih za VR. Slednje je upoštevano tudi pri velikosti mobilne naprave, in sicer očala verzije 2.0 omejujejo velikost mobilne naprave na 6 inčev. V primeru, da ima uporabnik mobilno napravo večjih dimenzij, lahko izdelava večja VR očala in brez težav uporablja aplikacijo [11].

3.2 Programska oprema

V tem poglavju so opisana orodja, okolja, prijemi in tehnologije, ki smo jih uporabljali za razvoj aplikacije.

3.2.1 Unity

Unity je razvojno okolje, ki ga je izdalo podjetje Unity Technologies **Napaka! Vira sklicevanja n i bilo mogoče najti..** Uporablja se za razvoj iger za osebne računalnike, igralne konzole, mobilne naprave in spletne strani. Prednost razvojnega okolja Unity je podpora velikega nabora ciljnih platform, kot so Android, iOS, Windows, Windows Phone 8, Linux, macOS, Wii, Wii U, Xbox 360, Xbox One, BlackBerry 10, PlayStation 4, PlayStation Vita, itd. [13]. Za kodiranje omogoča tri skriptne jezike in sicer JavaScript, C# in Boo. S programskim okoljem je povezan spletni portal AssetStore, kjer lahko razvijalci dostopajo do vsebin, ki so brezplačne ali plačljive. Prav tako lahko razvijalci na tem portalu delijo svoje vsebine. Vsebine na portalu predstavljajo celotne projekte ali pa gre le za dele projektov, kot so zvočni posnetki, objekti, ozadja, skripte, itd. Unity 3D ponuja tri plačljive in eno brezplačno licenčno izvedbo. Razlika med plačljivimi in brezplačno izvedbo je v tem, da imajo plačljive izvedbe več funkcionalnosti in boljšo podporo [14]. Za potrebe diplomskega dela je bila uporabljena 64 bitna brezplačna izvedba Unity Personal 5.5.0f3. Razvojno okolje Unity prikazuje Slika 3.3. Okolje je bilo nameščeno na 64 bitnem OS Windows 10 Profesional.



Slika 3.3: Razvojno okolje Unity

3.2.2 C#

C# je objektno orientirani programski jezik. Razvit je bil s strani podjetja Microsoft in deluje na platformi .NET. Sintaksa programskega jezika C# je podobna kot pri jezikih C, C++ in Java. Na podlagi izkušenj iz drugih programskih jezikov, so v programskem jeziku C# zajete njihove najboljše značilnosti in odpravljene njihove pomanjkljivosti. Programski jezik C# je definiran s CLI (Common Language Infrastructure) standardom, ki je v skladu z ISO (International Organization for Standardization) in ECMA (European Computer Manufacturers Association) standardi. Del kode napisane v programskem jeziku C# je enostavno povezati z deli kode, ki so napisani v drugih programskih jezikih razvitih na podlagi standarda CLI [15]. V diplomskem delu je bil uporabljen programski jezik C# v razvojnem orodju Visual Studio 2015, ki ga razvojno okolje Unity uporablja kot privzeto orodje za izvajanje skript, napisanih v programskem jeziku C#.

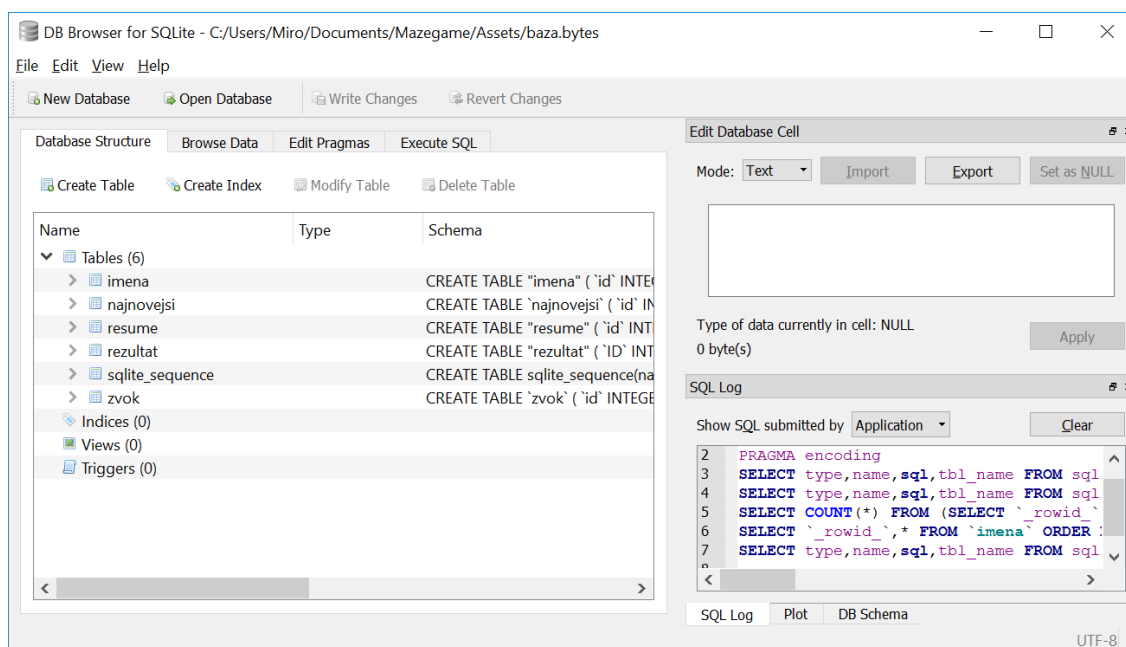
3.2.3 SQLite

SQLite je odprtokodna podatkovna baza, ki se v večini primerov uporablja za lokalno shranjevanje podatkov. Za razliko od drugih podatkovnih baz, SQLite ne potrebuje dodatnega podatkovnega strežnika za shranjevanje podatkov, saj je lahko sestavni del aplikacije. Podpira različne programske jezike in je prenosljiva med različnimi operacijskimi sistemi. Prednost

podatkovne baze SQLite je majhna poraba delovnega pomnilnika. Prav tako podatkovno bazo SQLite odlikuje majhna poraba prostora na lokalnem mediju. Ravno zaradi teh lastnosti je zelo priljubljena pri uporabi v spletnih brskalnikih in v aplikacijah za mobilne naprave[16]. V okviru diplomskega dela je bila uporabljena knjižnica SQLite Mono verzije 3.

3.2.4 DB Browser for SQLite

DB Browser for SQLite je odprtokodno orodje za delo s podatkovno bazo SQLite. Uporablja grafični vmesnik, ki omogoča izvajanje operacij kot so kreiranje PB, dodajanje tabel, urejanje podatkov v tabelah, itd. DB Browser for SQLite podpira OS Windows in MacOS ter večino distribucij operacijskega sistema Linux [17]. Za potrebe diplomskega dela je bila uporabljena verzija 3.9.1 programa Browser for SQLite. Browser for SQLite prikazuje Slika 3.4.

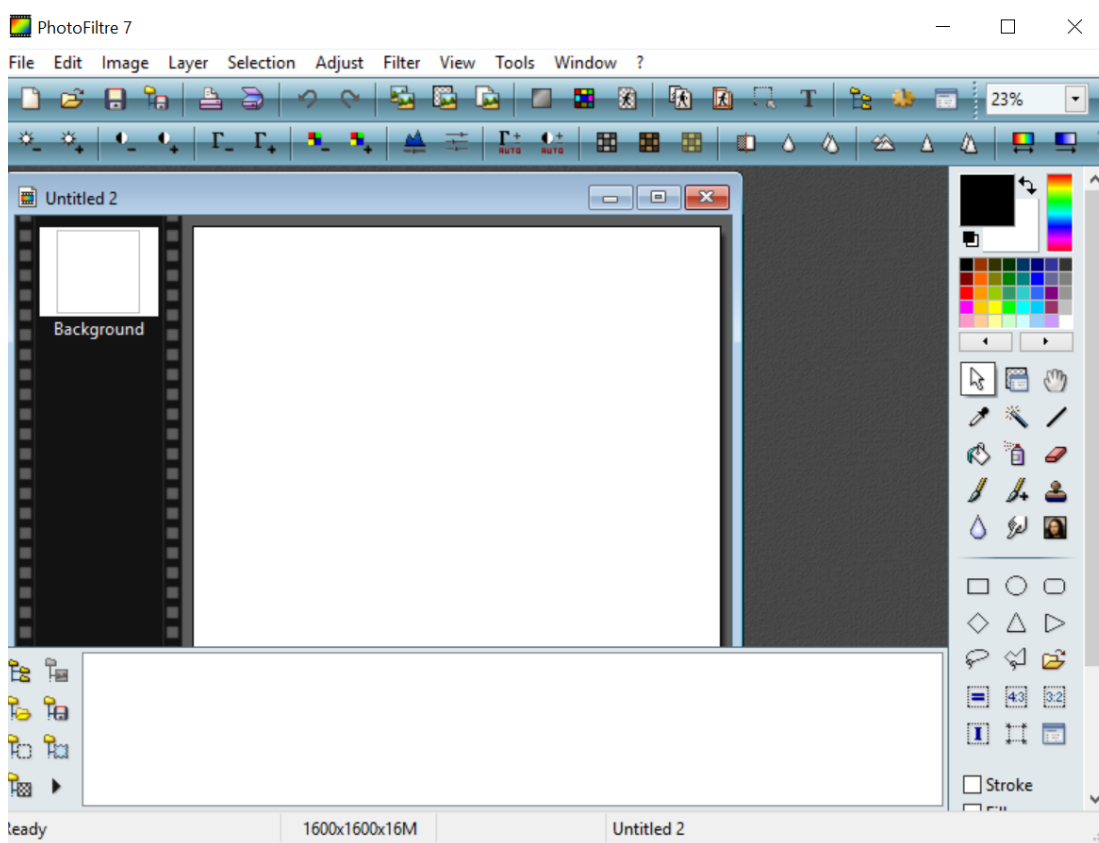


Slika 3.4: Browser for SQLite

3.2.5 PhotoFiltre

PhotoFiltre je enostavno programsko orodje za urejanje slik. Orodje omogoča vse osnovne operacije za urejevanje slik, torej obračanje, spreminjanje velikosti, dodajanje filtrov, shranjevanje v različnih formatih in kakovosti. Programsko orodje je brezplačno za osebne in izobraževalne namene, za komercialno ali profesionalno uporabo pa je potreben nakup licence

[18]. Za potrebe diplomskega dela je bila uporabljena verzija PhotoFiltre 7 v angleškem jeziku. PhotoFiltre 7 prikazuje Slika 3.5.



Slika 3.5: PhotoFiltre 7

Poglavje 4 Razvoj aplikacije

V okviru tega poglavja je opisan potek razvoja aplikacije. Predstavljeni so praktični primeri uporabe tehnologij in orodij opisanih v predhodnem poglavju.

4.1 Ideja

Razvijanje aplikacij za mobilne naprave že vrsto let ni nobena posebnost, kar pomeni, da danes obstaja že ogromno aplikacij za mobilne naprave, ki služijo za razne namene. Vsaka uvedba nove platforme je za razvijalce dobra priložnost, da razvijejo aplikacijo ki bo nekaj novega, bodisi da bo izkoristila novo tehnologijo mobilne naprave ali pa bo zgolj vsebinsko nekaj novega za to platformo. V večini primerov je uvedba nove platforme izziv za razvijalca, saj razbije monotonijo uporabe že znanih metod in orodij ter se preizkusi na še neraziskanem področju razvoja.

Ideja je bila razviti enostavno aplikacijo za operacijski sistem Android , ki bo uporabljala platformo Google Cardboard. Prav za to platformo smo se odločili, ker nam je platforma GC bila precej neznana iz razvojnega vidika. To je predstavljalo dodaten izziv in motivacijo za izdelavo aplikacije. Na začetku smo veliko razmišljali o tem, kakšno aplikacijo naj razvijemo. Ideja je dopuščala veliko fleksibilnosti glede realizacije, saj na začetku nismo točno določili kaj mora aplikacija vsebovati in kako bo izgledala. Po pregledu drugih dostopnih aplikacij za platformo GC smo se odločili, da bomo izdelali miselno igro, ki bo vsebovala dobre prakse in čim manj pomanjkljivosti.

4.2 Načrtovanje

Načrtovanja smo se lotili tako, da smo najprej opredelili pravila za miselno igro, torej samo vsebino. Na ta način smo ustvarili osnovo iz katere smo gradili celotno aplikacijo. Za primer smo preizkusili nekaj brezplačnih aplikacij v trgovini Play. Bili smo pozorni na to, da so igre razvite za platformo GC, saj so nas zanimale le te. Po pregledu obstoječih iger v trgovini Play smo naleteli na več pomanjkljivosti, ki so prisotne v nekaterih brezplačnih igrah. Te pomanjkljivosti so:

- takoj po zagonu aplikacije se odpre glavni del igre in se igra že začne;
- slabo izdelana navigacija v igri, če le ta vsebuje menije;
- navodila za upravljanje igre in opis cilja igre so pomanjkljivi ali pa jih sploh ni;
- igralec ni obveščen, kdaj mora mobilno napravo vstaviti v VR napravo;
- v igrah manjkajo rezultati;
- ni kontrole nad jakostjo zvoka.

Na podlagi preizkusov smo se odločili, da bomo v naši igri poskusili odpraviti večino omenjenih pomanjkljivosti, ki jih vsebujejo brezplačne igre v trgovini Play. Zaradi tega igra vsebuje 7 različnih zaslonov, poleg glavnega scenarija same igre. Aplikacija uporablja podatkovno bazo. V vseh scenarijih se podatkovna baza uporablja za branje, v večini pa tudi za pisanje različnih tipov podatkov. Kontrola nad zvokom je aktivna v scenarijih za katere smo predpostavljali, da je potrebna. Prav tako je v igri razvit scenarij v obliki animacije, ki uporabnika opozori, kdaj mora mobilno napravo vstaviti v VR napravo. Posebno pozornost smo posvetili navigaciji v igri, kjer je premikanje med meniji jasno označeno in intuitivno.

Izdelali smo miselno igro, ki ima obliko labirinta po katerem se igralec premika in išče izhod. Glavni del igre je izdelan v VR načinu. V labirint smo postavili objekte po katerih se igralec orientira. Prav tako smo dodali ovire, katere mora igralec premagati, v nasprotnem primeru izgubi igro. Drugi način za izgubo igre je ta, da igralcu zmanjka časa, ki je določen za iskanje izhoda iz labirinta. Tudi izpis izida je izdelan v VR načinu. Igro smo opremili s podatkovno bazo, ki shranjuje rezultate igralcev, ki so uspešno končali igro. Aplikacija vsebuje še zaslon za vnos imena, glavni meni, prikaz rezultatov in zaslon z animacijo, ki prikazuje spremembo načina prikaza aplikacije. Slednji zasloni niso narejeni v VR načinu, kar omogoča lažje upravljanje z njimi.

Okolje Unity v osnovni namestitvi ne vsebuje način prikaza za VR tehnologijo. Za tak način prikaza slike je bilo potrebno najprej uvoziti knjižnico GVR, ki smo jo našli na uradni strani Google Cardboard za razvijalce.

Svetlobo v aplikaciji smo določili tako, da je osvetlila celoten prostor, ki ga uporabnik uporablja pri igranju igre.

4.3.2 2D oblikovanje

Za ostale zaslone v aplikaciji, na katerih so prikazani meniji, navodila, rezultati, animacija in vnosno polje za vnos imena, smo uporabili 2D (two dimensional) način prikaza. Odločitev je bila sprejeta na podlagi testiranj pri katerih smo ugotovili, da tak način prikaza predstavlja boljše uporabniško izkušnjo. Prikaz je bolj jasen, izbiranje pa posledično hitrejše. Primer 2D načina prikaza prikazuje Slika 4.2.

Grafično oblikovanje v 2D načinu je bilo bolj enostavno. Uporabljali smo gradnike, ki jih ponuja razvojno okolje Unity. Ti gradniki so vnosno polje, tekstovno polje, gumb in slika.



Slika 4.2: Načrtovanje pomožnih delov aplikacije

4.4 Zvok

Pri razvoju aplikacije smo uporabili različne formate zvočnih datotek, ki smo jih uvozili v razvojno okolje Unity. Primeri uporabljenih končnic zvočnih datotek so wav, mp3, aif in ogg.

Zvok v aplikaciji lahko razvrstimo v dve večji skupini, in sicer zvoke ozadja in zvočne efekte, ki se začnejo predvajati ob sprožitvi povezanega elementa. V večini primerov je sprožitev zasnovana tako, da uporabnik izbere opcijo iz seznama ali se dotakne objekta na katerega je pripet zvočni posnetek.

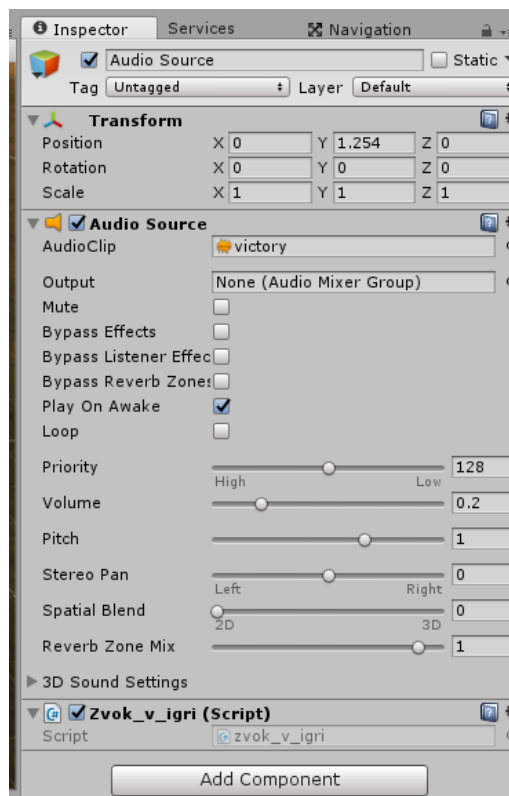
Zvok ozadja je prisoten skozi večji del aplikacije. V menijih je zvok ozadja nastavljen tako, da se ob preklopu menijev nadaljuje, kjer je pri prejšnjem meniju končal. Izjema je zvok ozadja v samem igralnem delu, saj je le ta drugačen od zvoka menijev, ki igra v ozadju skozi celotno aplikacijo.

V igralnem delu aplikacije smo dodali kopico zvočnih efektov. Primeri zvočnih efektov pri igranju igre so, zvočni efekt korakov, zvočni efekt ovir, zvočni efekt ob končani igri in zvočni efekt ob izteku časa. Tudi v menije smo dodali zvočne efekte, ki se v večini primerov sprožijo ob izbiri opcije v meniju.

Večji del aplikacije omogoča kontrolo nad zvokom na način, da lahko zvok vklopimo ali izklopimo. Stanje se odraža na nivoju cele aplikacije. Vsaka nastavitev zvoka se beleži v lokalno podatkovno bazo, kar pomeni da se tudi po izhodu iz aplikacije obdrži in se ob ponovnem zagonu aplikacije nastavi na vrednost, ki je veljala ob izhodu iz aplikacije. Razvojno okolje Unity ponuja urejanje nastavitvev zvočnih datotek v grafičnem vmesniku. Nastavitve se lahko urejajo tudi preko skript. Oboje prikazujeta sliki Slika 4.3 in Slika 4.4. Pri razvoju aplikacije smo uporabili oba načina urejanja nastavitvev zvočnih datotek.

```
void Start () {  
    audio = GetComponent();  
  
    if (get() == "0")  
    {  
        audio.mute = true;  
    }  
    else  
    {  
        audio.mute = false;  
    }  
}
```

Slika 4.3 Urejanje zvočnih nastavitvev s pomočjo skripte



Slika 4.4 Urejanje zvočnih nastavitev v grafičnem vmesniku razvojnega okolja Unity

4.5 Kodiranje

Aplikacija vsebuje funkcionalnosti, ki jih kontroliramo s pomočjo skript. Napisane so v programskem jeziku C#. Uporabili smo večje število krajših skript, ker je aplikacija precej obsežna in bi v nasprotnem primeru morali uporabljati slabe prakse programiranja. Vse funkcionalnosti aplikacije so zajete v približno 40. skriptah.

Vsaka skripta, ki jo kreiramo v okolju Unity ima podobno začetno zgradbo. Sestavljata jo dva dela, ki sta implementirana v obliki funkcij brez vsebine. Prvi del predstavlja funkcija, ki se imenuje Start. Primarni namen funkcije je inicializacija spremenljivk, lahko pa dodamo tudi kodo, če želimo, da se bo izvedla ob zagonu programa. Za slednjo možnost je bolj pravilno, da uporabimo funkcijo Awake. Drugi del je funkcija, ki se imenuje Update. Kliče se na vsak okvir (ang. frame), zato na tem delu programa ni priporočljivo operirati s podatkovno bazo. Kodiranje v tej funkciji se uporablja predvsem za sprotno preverjanje ali nastavljanje vrednosti spremenljivkam. Začetno zgradbo skripte prikazuje Slika 4.5.


```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class testna : MonoBehaviour {

    // Use this for initialization
    void Start () {

    }
    void

    // Update is called once per frame
    void Update () {

    }

}

```

Slika 4.5: Začetna zgradba skripte

V skriptah se pojavljajo podobne funkcije, saj nekatere skripte izvajajo podobne operacije. Primer take operacije je povezava na podatkovno bazo. Takšno funkcijo vsebuje večina skript, se je pri vsakem operiranju s podatkovno potrebno odpreti povezavo. Funkcijo za povezovanje s podatkovno bazo prikazuje Slika 4.6.

```

private string conn()
{
    try
    {
        if (Application.platform != RuntimePlatform.Android)
        {
            path = Application.dataPath + "/baza.bytes";
        }
        else
        {
            path = Application.persistentDataPath + "/baza.bytes";
            if (!File.Exists(path))
            {
                WWW load = new WWW("jar:file://" + Application.dataPath + "!/assets/" + "baza.bytes");
                while (!load.isDone)
                {
                }
                File.WriteAllBytes(path, load.bytes);
            }
        }
    }
    catch (Exception ex)
    {
        Debug.Log(ex.ToString());
    }
    return path;
}

```

Slika 4.6: Funkcija za povezovanje s podatkovno bazo

Za implementacijo funkcionalnosti na menijih smo uporabljali bolj enostavne funkcije. Primeri takšnih funkcij so odzivi na dogodke, kot so izbor gumba ali vpis teksta v tekstovno polje. Prav tako smo s pomočjo skript implementirali funkcije, ki sprožijo izjeme v primeru zaznave

nepravilnosti v aplikaciji. Največ nepravilnosti smo omejili na delu aplikacije, kjer uporabnik vpiše svoje ime. Primeri teh nepravilnosti so prazno vnosno polje, predolgo ime in brisanje praznega vnosnega polja.

Implementacija funkcionalnosti v samem igralnem delu aplikacije je bila nekoliko bolj specifična, saj je bilo treba implementacijo prilagoditi 3D svetu. Premikanje igralca po prostoru je eden takih primerov. V funkciji je določena hitrost in smer gibanja igralca. V primeru, da igralec pogleda proti tlam, se gibanje ustavi. Implementirana je tudi kontrola zvočnega efekta korakov, saj se zvočni efekt za hojo pri zaustavitvi igralčevega gibanja izklopi. Funkcijo za premikanje igralca v prostoru prikazuje Slika 4.7.

```
void Update() {
    if (walking)
    {
        transform.position = transform.position + Camera.main.transform.forward * 2f * Time.deltaTime;
    }

    Ray ray = Camera.main.ViewportPointToRay(new Vector3(.5f, .5f, 0));
    RaycastHit hit;

    if (Physics.Raycast(ray, out hit)){

        if (hit.collider.name.Contains("Plane")){

            walking = false;
            hoja.mute = true;
        }
        else
        {
            walking = true;
            if (zvok_hoja == "1")
            {
                hoja.mute = false;
            }
        }
    }
}
```

Slika 4.7: Funkcija za premikanje igralca v prostoru

4.6 Podatkovna baza

4.6.1 Opis podatkovne baze

Aplikacija uporablja lokalno podatkovno bazo. Implementirana je s pomočjo knjižnice SQLite, ki jo osnovna namestitev okolja Unity, ne vsebuje. Potrebno je bilo uvoziti datoteke, ki smo jih našli na uradni spletni strani SQLite.

Podatkovno bazo smo morali najprej kreirati. To smo izvedli s pomočjo orodja DB Browser for Sqlite. Uporabili smo podatkovno bazo s končnico bytes.

Sprva smo nameravali lokalno podatkovno bazo uporabljati samo za beleženje rezultatov, ki jih dosežejo igralci. V okviru procesa razvoja pa se je izkazalo, da je smiselno lokalno podatkovno bazo uporabiti še za druge namene. Primer takšne uporabe je beleženje nastavitve zvoka, kjer smo se izognili globalnim spremenljivkam pri kodiranju in obdržali nastavev zvoka tudi pri izhodu iz aplikacije.

V podatkovni bazi z imenom baza.bytes smo kreirali naslednjih 5 tabel:

- Imena: hrani podatek o imenu trenutnega igralca;
- Najnovejši: hrani podatke o imenu in doseženem času igralca, ki je nazadnje igral igro;
- Resume: hrani podatke o imenu, poziciji, času in statusu shranjene igre igralca, ki je nazadnje igral igro;
- Rezultat: hrani rezultate igralcev, ki so uspešno zaključili igro. Rezultati so shranjeni v obliki imena in doseženega časa;
- Zvok: hrani podatke o nastavitvi zvoka v igri. Strukturo podatkovne baze prikazuje Slika 4.8.

▼ imena		CREATE TABLE "imena" ("id" INTEGER NOT NULL, "ime" TEXT)
id	INTEGER	"id" INTEGER NOT NULL
ime	TEXT	"ime" TEXT
▼ najnovejši		CREATE TABLE "najnovejsi" ("id" INTEGER NOT NULL, "cas" INTEGER, "ime" TEXT, PRIMARY KEY("id"))
id	INTEGER	"id" INTEGER NOT NULL
cas	INTEGER	"cas" INTEGER
ime	TEXT	"ime" TEXT
▼ resume		CREATE TABLE "resume" ("id" INTEGER NOT NULL, "name" TEXT, "pozicijax" REAL, "pozicijaz" REAL, "time" INTEGER, "flag" INTEGER)
id	INTEGER	"id" INTEGER NOT NULL
name	TEXT	"name" TEXT
pozicijax	REAL	"pozicijax" REAL
pozicijaz	REAL	"pozicijaz" REAL
time	INTEGER	"time" INTEGER
flag	INTEGER	"flag" INTEGER
▼ rezultat		CREATE TABLE "rezultat" ("ID" INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT, "cas" NUMERIC NOT NULL, "ime" TEXT)
ID	INTEGER	"ID" INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT
cas	NUMERIC	"cas" NUMERIC NOT NULL
ime	TEXT	"ime" TEXT
▼ zvok		CREATE TABLE "zvok" ("id" INTEGER NOT NULL, "zvok" INTEGER NOT NULL, PRIMARY KEY("id"))
id	INTEGER	"id" INTEGER NOT NULL
zvok	INTEGER	"zvok" INTEGER NOT NULL

Slika 4.8: Struktura podatkovne baze Baza.bytes

Izvajanje operacij nad bazo je bilo izvedeno s pomočjo skript v programskem jeziku C#. Najbolj pogoste operacije nad bazo so bile v obliki stavkov select, insert in update. Za izvajanje operacij nad bazo je bilo potrebno najprej vzpostaviti povezavo z bazo. V metodi za vzpostavitev povezave je definirana relativna pot do podatkovne baze, ki je posebej določena za platformo Android. Za odpiranje povezave s podatkovno bazo smo klicali metodo za vzpostavitev

povezave s podatkovno bazo. Pri kodiranju smo morali biti pozorni na zapiranje podatkovne baze, saj več skript operira z isto podatkovno bazo.

4.6.2 Branje iz podatkovne baze

Branje iz podatkovne baze smo implementirali s pomočjo stavka `select`, kar prikazuje Slika 4.9. Pri tem smo bili pozorni, da branja iz baze ne implementiramo v funkciji `update`, ki se kliče enkrat na okvir. V nasprotnem primeru se lahko zgodi preobremenitev mobilne naprave in posledično neodzivnost celotne aplikacije. Enako pravilo velja tudi za pisanje v bazo.

```
private string get()
{
    condb = new SqlConnection("URI=file:" + conn());
    condb.Open();
    string rez = "";
    cmd_db = new SqlCommand("select * from Zvok", condb);
    rdr = cmd_db.ExecuteReader();
    while (rdr.Read())
    {
        rez = rdr[0].ToString();
    }
    disconnect();

    return rez;
}
```

Slika 4.9: Branje iz podatkovne baze

4.6.3 Pisanje v podatkovno bazo

Pisanje v podatkovno bazo je v večini primerov sestavljal stavek `update`. Na tak način smo zamenjali obstoječe vrednosti z novimi vrednostmi. To nam je olajšalo kodiranje, saj smo določene vrednosti želeli videti in nastavljeni v ločenih skriptah. Prednost posodabljanja vrednosti je ohranitev zelo majhne velikosti podatkovne baze, kar je za platformo Android zelo priporočljivo. Pri mobilnih aplikacijah po navadi ne želimo podatkovne baze, ki bi sčasoma zasedla ogromno pomnilnika. Primer posodabljanja vrednosti v bazi je posodabljanje nastavitve vklopa in izklopa zvoka, kjer ne potrebujemo podatkov preteklih nastavitvev, kaj prikazuje Slika 4.10.

```
private void setValue(string vrednost)
{
    condb = new SqlConnection("URI=file:" + conn());
    condb.Open();

    if (vrednost.Equals("false"))
    {
        cmd_db = new SqlCommand
        ("UPDATE Zvok SET zvok = 1 WHERE id = 1", condb);
        cmd_db.ExecuteNonQuery();
    }
    else
    {
        cmd_db = new SqlCommand
        ("UPDATE Zvok SET zvok = 0 WHERE id = 1", condb);
        cmd_db.ExecuteNonQuery();
    }
    disconnect();
}
```

Slika 4.10: Posodabljanje vrednosti v podatkovni bazi

V podatkovni bazi smo dodajali zapise samo v tabelo Rezultat, saj smo beležili rezultate in imena vseh igralcev, ki so uspešno zaključili igro. Podatke, ki smo jih želeli dodati v podatkovno bazo smo najprej obdelali v programskem jeziku C#. Dodajanje podatkov v podatkovno bazo smo izvedli s pomočjo stavka insert, kar prikazuje Slika 4.11.

```
private void vpisiRezultat(int cas)
{
    condb = new SqlConnection("URI=file:" + conn());
    condb.Open();
    string ime = pridobiime();
    cmd_db = new SqlCommand
    ("INSERT into rezultat (cas,ime) VALUES ('+cas+', '" + ime + "')", condb);
    cmd_db.ExecuteNonQuery();
    disconnect();
}
```

Slika 4.11: Vstavljanje v PB

Poglavje 5 Delovanje aplikacije

V tem poglavju so sprva na kratko opisane ključne lastnosti aplikacije. V nadaljevanju je podrobno opisano delovanje aplikacije. Delovanje je razdeljeno na tri dele, ki se med seboj razlikujejo po načinu prikaza in namenu uporabe.

5.1 Osnovne lastnosti aplikacije

Aplikacija uporablja angleški jezik za komunikacijo z uporabnikom. Za angleščino smo se odločili, ker je najbolj razširjen jezik in je večja verjetnost, da ga bo uporabnik razumel, kot če bi uporabili kateri drugi jezik. Za aplikacijo je s strani razvijalca določena lega naprave, ki je ležeče levo (ang. Landscape left) usmerjena. Namen te lege je pravilen prikaz pri izvajanju aplikacije v VR načinu. Prav tako je določena tudi shramba nameščene aplikacije, ki je notranja shramba (ang. Internal storage) mobilne naprave. Primaren namen te vrste shrambe na mobilni napravi je delovanje na mobilnih napravah, ki ne podpirajo zunanje shrambe. Sekundarni namen pa je hitrejše izvajanje same aplikacije. Velikost namestitvene datoteke aplikacije s končnico apk znaša 47 MB.

5.2 Zagon aplikacije

Aplikacijo je najprej potrebno izvoziti iz okolja Unity. Pri pravilnem izvozu za platformo Android je aplikacija skupaj s podatkovno bazo, strnjena v izvedljivo datoteko s končnico apk. Takšna datoteka je pripravljena za namestitev aplikacije na mobilno napravo z OS Android.

Po namestitvi je aplikacija prikazana na seznamu ikon v mobilni napravi. Oblika ikone je izdelana s strani razvijalca aplikacije, kot prikazuje Slika 5.1.



Slika 5.1: Ikona aplikacije

Aplikacija, vidna med ikonami, je pripravljena na zagon. Nekaj sekund po zagonu se prikaže logotip razvojnega okolja v katerem je aplikacija izdelana. Prikaz logotipa traja približno 3 sekunde in ni določen s strani razvijalca aplikacije, temveč je privzet zaslon okolja v katerem je aplikacija izdelana, kar prikazuje Slika 5.2.



Slika 5.2: Logotip razvojnega okolja

Po prikazu logotipa, se uvodni del aplikacije nadaljuje s prikazom logotipa igre, ki je bil razvit s strani razvijalca, kot to prikazuje Slika 5.3. Prav tako je bilo s strani razvijalca določeno trajanje prikaza logotipa aplikacije. Čas smo nastavili ne 5 s.



Slika 5.3: Logotip aplikacije

5.3 Uporabniški vmesnik

V tem podpoglavju so zajeti vsi zasloni, ki uporabljajo 2D način prikaza. To so v večini primerov meniji in deli, kjer je bilo smiselno uporabiti 2D način prikaza. V okvir uporabniškega vmesnika spadajo deli aplikacije, kjer uporabnik še ni vstavil mobilne naprave v VR napravo.

5.3.1 Vnos imena

Po uvodnem delu aplikacije sledi prvi del aplikacije, kjer je aktivno vključen uporabnik. Aplikacija na tem delu začne uporabljati podatkovno bazo, saj iz nje pridobi podatke ali je pri zadnjem izhodu bil zvok vklopljen in na podlagi tega podatka nastavi vrednost zvoka. Prav tako s pomočjo podatkovne baze pridobi podatke o tem, ali obstaja na napravi nedokončana igra. Igra se šteje za nedokončano v primeru, da uporabnik zapre aplikacijo med igro. Na ta način se v podatkovno bazo shrani ime, čas in pozicija uporabnika. V primeru shranjene igre aplikacija na prvem zaslonu ponudi uporabniku dve izbiri ki sta: nadaljevanje zadnje igre ali vnos drugega imena. Z vnosom drugega imena se shranjena igra izbriše iz podatkovne baze ter se začne nova igra. Zaslona za vnos imena v primeru obstoja shranjene igre prikazuje Slika 5.4.



Slika 5.4: Zaslona za vnos imena v primeru obstoja shranjene igre

V primeru, da shranjena igra ne obstaja, se uporabniku prikaže zaslon za vnos novega imena. Enako se zgodi, če shranjena igra obstaja, uporabnik pa izbere možnost vnosa novega imena. Zaslona za vnos imena ima tudi gumb zgoraj levo, s katerim lahko vklopimo ali izklopimo zvok. Na zaslonu zgoraj desno je gumb, ki ob sprožitvi zapre aplikacijo. Izhod iz aplikacije lahko dosežemo tudi ob sprožitvi fizične tipke nazaj, na mobilni napravi. Na sredini zaslona je tipkovnica, ki omogoča vnos velikih in majhnih črk, namenjena je vnosu imena. Na spodnji desni strani tipkovnice je gumb enter, s katerim potrdimo vnos imena. Pod tipkovnico se nahaja tekstovno polje, na katerem se v primeru napak izpiše opozorilo, kot kaže Slika 5.5. Možno je sprožiti izpis treh različnih tipov napak in sicer:

- Name is required: ko hočemo potrditi vnos imena s praznim nizom;
- Name field is empty: ko želimo brisati črke v levo pri praznem nizu;
- Name is too long: ko želimo vnesti ime daljše od 15 znakov.



Slika 5.5: Osnovni zaslon za vnos imena

5.3.2 Glavni meni

Po potrditvi vpisanega imena nam aplikacija ponudi glavni meni. Zvok v glavnem meniju se nadaljuje iz prejšnjega menija. V primeru, da smo na prejšnjem zaslonu zvok utišali, je tudi v glavnem meniju utišan. Na vrhu zaslona je prikazan pozdrav, ki vsebuje ime vpisano v prejšnjem zaslonu. Glavni meni vsebuje enaka gumba za zvok in izhod kot v prvem delu aplikacije, funkcionalnosti gumbov sta prav tako enaki. Funkcionalnost tipke nazaj na mobilni napravi je enaka kot pri prejšnjem zaslonu za vnos imena. Poleg omenjenih dveh gumbov, zaslon glavnega menija vsebuje še en gumb v spodnjem desnem kotu. Funkcionalnost tega gumba je prikaz zaslona z navodili igre.

Na sredini ekrana so prikazane tri opcije. Zgornja opcija služi zagonu nove igre. Srednja opcija je namenjena nadaljevanju shranjene igre, ko shranjena igra ne obstaja je gumb neaktiven. Spodnja opcija je namenjena prikazu rezultatov vseh igralcev, ki so igro uspešno zaključili. Glavni meni prikazuje Slika 5.6.



Slika 5.6: Glavni meni

5.3.3 Rezultati

Na zaslonu, ki prikazuje rezultate, sta enaka gumba za zvok in izhod kot pri glavnem meniju, prav tako imata enake funkcionalnosti. Spremeni se funkcionalnost tipke nazaj na mobilni napravi, saj se ob izboru tipke prikaže glavni meni. Na dnu ekrana leži gumb reset, ki pobriše vse rezultate. V spodnjem levem kotu se nahaja gumb, ki vrne uporabnika nazaj na glavni meni, kar prikazuje tudi Slika 5.7.



Slika 5.7: Rezultati

Rezultati so generirani na podlagi uspešno končanih iger. Za uspešno končano igro se šteje vsaka igra, kjer igralec najde izhod iz labirinta. Pri tem se v podatkovno bazo shrani ime in dosežen čas. Pri prikazu rezultatov skripta pridobi podatke iz podatkovne baze. Urejeni so po doseženem času, v naraščajočem zaporedju.

5.3.4 Navodila

Do navodil igre je urejen dostop preko glavnega menija, kjer izberemo gumb s simbolom vprašaja v spodnjem desnem kotu. Zaslon z navodili vsebuje tri gumbe in sicer gumb za zvok, za izhod in za vrnitev na glavni meni. Razporeditev teh gumbov in njihovih funkcionalnosti sta enaki kot pri zaslonu z rezultati. Tipka nazaj na mobilni napravi ima enako funkcionalnost, kot na zaslonu z rezultati, torej vrnitev na glavni meni. Zvok v rezultatih se nadaljuje iz glavnega menija, prav tako tudi kontrola zvoka.

Vsebina zaslona z navodili vsebuje kratek opis upravljanja igralca v igri, na kaj mora biti pozoren in kaj je cilj igre. Kot že omenjeno, so navodila napisana v angleškem jeziku in so skržena v kratko in jasno obliko z namenom razumevanja uporabnika, ki pozna zgolj osnove jezika. Zaslon z navodili igre prikazuje Slika 5.8.

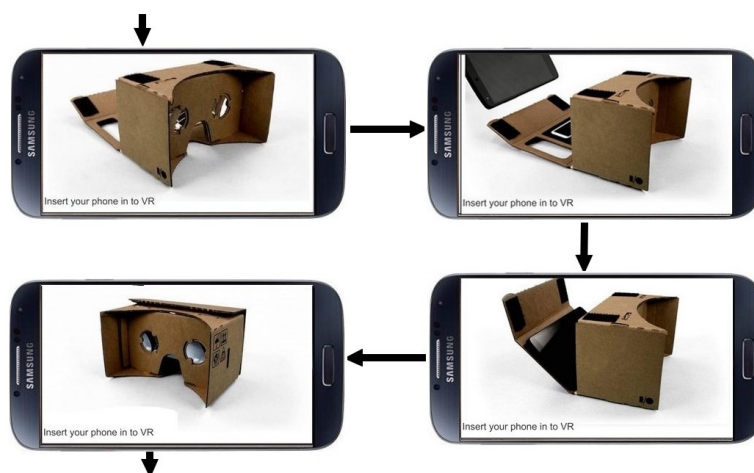


Slika 5.8: Zaslon z navodili igre

5.3.5 Animacija

Animacija je del uporabniškega vmesnika, ki se pojavi, ko je mobilno napravo potrebno vstaviti v VR napravo, torej ko v glavnem meniju izberemo nova igra ali nadaljevanje shranjene igre. Sestavljajo jo štirje različni zasloni, ki v pravilnem zaporedju simbolizirajo postopek vstavljanja mobilne naprave v VR napravo, kar prikazuje Slika 5.9. Trajanje animacije je določeno na podlagi meritev časa, ki ga uporabnik potrebuje za vstavljanje mobilne naprave v VR napravo. Meritve so pokazale, da je optimalen čas za ta postopek približno 5 s. V aplikaciji je nastavljen čas trajanja animacije na 7 sekund, saj je upoštevan zamik nepoznavanja aplikacije s strani uporabnika.

Ta del aplikacije ne vsebuje zvoka in drugih elementov, temveč samo prikaz postopka vstavitve mobilne naprave v VR napravo. Edina kontrola je tipka nazaj na mobilni napravi, ki uporabnika vrne v glavni meni.



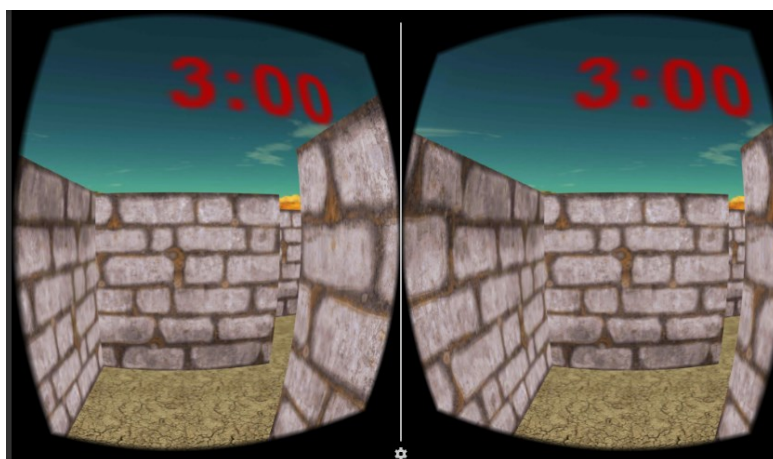
Slika 5.9 : Potek animacije

5.4 Igralni del aplikacije

V opisu igralnega dela aplikacije je zajet opis igre in vseh zaslonov, ki so neposredno povezani z igro. Izdelan je v 3D načinu prikaza, z uporabo platforme GVR. Za pravilen prikaz igralnega dela aplikacije je potrebno mobilno napravo vstaviti v GVR napravo.

5.4.1 Igra

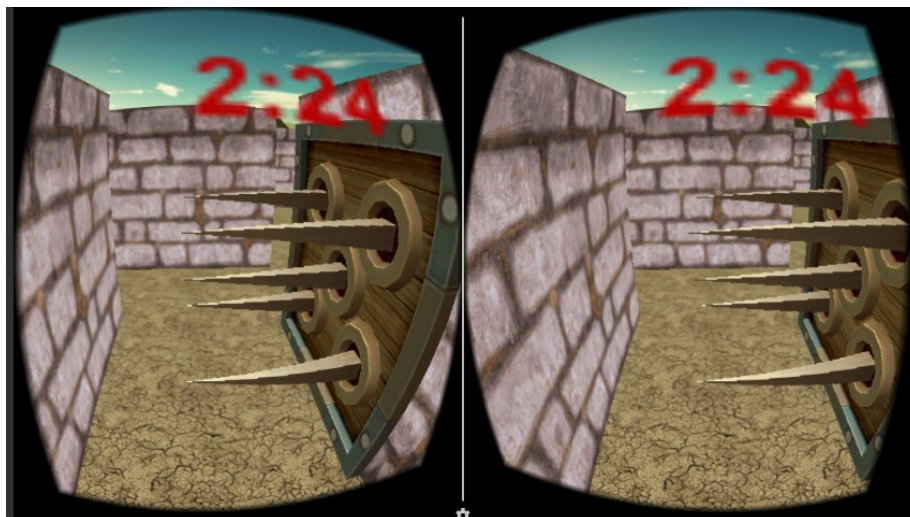
Igra predstavlja osrednji del aplikacije. Do igre uporabnik lahko dostopa preko glavnega menija z izborom opcije nova igra (ang. new game) ali nadaljevanje igre (ang. resume game). Uporabniku se prikaže animacija, ki ga opozarja na zamenjavo načina prikaza in nazorno predstavi, kaj mora uporabnik storiti v tem primeru. Po nekaj sekundah prikaza animacije se uporabniku prikaže igra, kar prikazuje Slika 5.10.



Slika 5.10: Igralni del aplikacije

Igra je sestavljena iz prostora v katerem se igralec premika. Premika se s konstantno hitrostjo. V primeru, da uporabnik usmeri pogled proti tlorisu, se gibanje igralca ustavi. Spremembo smeri gibanja lahko uporabnik določi s spremembo smeri VR naprave po horizontalni osi prostora.

Po prostoru so razporejeni 3D objekti, ki služijo lažji orientaciji uporabnika. Prav tako so po prostoru razporejene animirane ovire. Zgrajene so iz statičnega in animiranega dela. Pri večini je statični del pritrjen na steno prostora, medtem ko animirani del predstavlja orožje, ki v nekaterih fazah animacije igralcu zapira pot. V tem primeru mora igralec počakati, da se animacija izvede do faze, ko prepreka ne zapira več poti. V nasprotnem primeru se igralec dotakne ostrega dela prepreke in konča igro z negativnim izidom, kar prikazuje Slika 5.11.



Slika 5.11: Animirana prepreka v igri

V desnem kotu igralnega zaslona je odštevalnik, ki je nastavljen na začetno vrednost treh minut. Cilj igre je, da igralec poišče pot iz labirinta v času določenem na odštevalniku. Čas smo določili na podlagi testiranja uporabnikov, ki so prvič igrali igro. Testiranje je pokazalo, da so 3 minute dovolj za večino uporabnikov. Ko vrednost na odštevalniku pade pod 10 sekund, se aktivira zvočni efekt, ki uporabnika opozarja na iztekanje časa.

Izhod iz igralnega dela aplikacije med igranjem je možen na dva načina. Prvi način je izhod iz igralnega dela v glavni meni. Uporabnik ga doseže ob izbiri tipke nazaj na mobilni napravi. Drugi način omogoča izhod iz celotne aplikacije. Uporabnik ga doseže tako, da izbere tipko domov na mobilni napravi. V obeh primerih se stanje igre shrani v lokalno podatkovno bazo in ob ponovnem zagonu omogoča nadaljevanje igre.

5.4.2 Izid igre

Uporabnik lahko konča igro s pozitivnim ali negativnim izidom. V obeh primerih je uporabljen VR način prikaza izida igre. Zaslona se razlikujeta v tem, da se glede na izid izpiše drugačno sporočilo za uporabnika na vrhu zaslona. V primeru, da uporabnik konča igro s pozitivnim izidom, se poleg izida izpiše še mesto, ki ga je uporabnik zasedel, glede na čas v katerem je zaključil igro. Razlikujeta se tudi po zvočnem efektu, ki se predvaja ob zaključku igre. Zaslona za izid igre sta si podobna v uporabi menija, ki vsebuje tri izbire:

- play again (ponovi igro),
- main menu (izhod iz igre v glavni meni),

- exit game (izhod iz aplikacije).

Izbiri željene opcije uporabnik doseže tako, da se s kurzorjem premakne na opcijo in počaka, da se izbor pobarva. Če se uporabnik odloči za izbor druge opcije, lahko premakne kurzor z začetnega izbora, preden se le ta v celoti pobarva, kar prikazuje Slika 5.12. Tipke na mobilni napravi so v tem delu aplikacije onemogočene, nadomestile so jih izbire v meniju.



Slika 5.12: Zaslon ob zaključku igre

Poglavje 6 Sklepne ugotovitve

V diplomskem delu je predstavljena navidezna resničnost in njen pomen v sodobnem času. Prav tako so predstavljene najbolj znane naprave za prikaz navidezne resničnosti. Z namenom predstavitve primera uporabe navidezne resničnosti, je bila razvita aplikacija za GVR platformo. Aplikacijo, ki smo jo razvili, predstavlja miselna igra za platformo GVR in OS Android. Cilj naloge je bil razviti kratko miselno igro, ki bo vsebovala dobre razvijalske prakse in bo imela odpravljene pomanjkljivosti ostalih brezplačnih iger v trgovini Play. To smo dosegli z razvojem uporabniškega vmesnika in uporabo lokalne podatkovne baze. Uporabniško izkušnjo smo izboljšali z jasnim in intuitivnim uporabniškim vmesnikom, ki je primeren za komunikacijo z uporabniki različnih starosti.

Razvito igro bi lahko izboljšali tako, da bi v njej uporabljene texture in objekte bolj grafično dovršili. Vsebinsko pa bi igro lahko nadgradili z naključnimi ovirami ali celo z naključno gradnjo prostora. Poleg tega bi lahko v igro dodali več različnih prostorov ali večji prostor v katerem bi igralec iskal izhod dlje časa. Izdelano igro bi lahko tudi prilagodili za delovanje na drugih mobilnih sistemih, kot so iOS ali BlackBerry OS.

Aplikacija je bila preizkušena s strani uporabnikov različnih starosti. Odzivi so bili v večini primerov pozitivni, saj se nekateri uporabniki še niso srečali s platformo GVR in jim je ta način igranja predstavljal nekaj povsem novega.

Literatura

- [1] Wikipedia. Navidezna_resničnost [Online]. Dosegljivo 8.2.2017 na: https://sl.wikipedia.org/wiki/Navidezna_resničnost.
- [2] S. G. Weinbaum. (7. oktober 2007) Pygmalion's Spectacles [Online] EBook #22893. Dosegljivo 8.2.2016 na: <http://www.gutenberg.org/ebooks/22893>.
- [3] N. Foreman, L. Korallo (2014). PAST AND FUTURE APPLICATIONS OF 3-D (VIRTUAL REALITY) TECHNOLOGY. Scientific and Technical Journal of Information Technologies, Mechanics and Optics [Online].Dosegljivo 8.2.2017 na: <http://ntv.ifmo.ru/file/article/11182.pdf>
- [4] Proteus VR Labs LTD. Time Travel Through Virtual Reality [Online] Dosegljivo 8.2.2017 na: <https://www.freeflyvr.com/time-travel-through-virtual-reality/>
- [5] E. Gera. (July 2013). Vendetta Online takes its digital galaxies to the Oculus Rift. EDT [Online]. Dosegljivo 8.2.2016 na: <http://www.polygon.com/2013/7/24/4552076/vendetta-online-takes-its-digital-galaxies-to-oculus-rift>
- [6] Jae C. Hong. (Marec 2014). Facebook to buy Oculus virtual reality firm for \$2B. Associated Press [Online]. Dosegljivo 8.2.2016: <http://www.cbc.ca/news/technology/facebook-to-buy-oculus-virtual-reality-firm-for-2b-1.2586318>
- [7] Oculus Rift VR, LLC. (2017) Oculus Rift Overview [Online]. Dosegljivo 8.2.2017 na: <https://www3.oculus.com/en-us/rift/>
- [8] N. Ralph. (2015) . Samsung gear VR (2015) review [Online]. Dosegljivo 8.2.2017 na: <https://www.cnet.com/products/samsung-gear-vr-2015/>
- [9] D. Pierce. (16.4.2016) Inside Google's plan to make VR amazing for absolutely, positively, everyone [Online]. Dosegljivo 8.2.2017 na: <https://www.wired.com/2016/04/google-vr-clay-bavor/>

- [10] M. Schirmer, H.hopfner. (April 2009). Smartphone hardware sensors. Mobile media group [Online]. Dosegljivo 8.2.2016 na: <https://www.uni-weimar.de/kunst-und-gestaltung/wiki/images/Zeitmaschinen-smartphonesensors.pdf>
- [11] Copyright © BrizTechVR. (2015). Google Cardboard' Version 1 and 2 Comparison [Online]. Dosegljivo 8.2.2017 na: <https://vr.briztech.co.uk/cardboard-comparison.html>
- [12] J. Riccitiello. (23.10.2014). John Riccitiello sets out to identify the engine of growth for Unity Technologies. DENTAKI [Online] Dosegljivo 8.2.2017 na: <http://venturebeat.com/2014/10/23/john-riccitiello-sets-out-to-identify-the-engine-of-growth-for-unity-technologies-interview/>
- [13] ©Unity Technologies. (2017). Build once, deploy anywhere [Online]. Dosegljivo 8.2.2017 na: <https://unity3d.com/unity/multiplatform/>
- [14] ©Unity Technologies. (2017). Unity Store [Online]. Dosegljivo 8.2.2017 na: <https://store.unity.com/>
- [15] ECMA International, C# Language Specification. Švica. Standard ECMA-334 4-th edition, Junij 2006. [Online]. Dosegljivo 8.2.2016 na: <http://www.ecma-international.org/publications/files/ECMA-ST/Ecma-334.pdf>
- [16] (2017) About SQLite [Online] Dosegljiv 8.2.2017 na: <https://sqlite.org/>
- [17] J. Long. (2016). DB Browser for SQLite [Online] Dosegljivo: <http://sqlitebrowser.org/>
- [18] J. Harej. (1.4.2014) PhotoFiltre [Online]. Dosegljivo 8.2.017 na: <http://podpora.sio.si/photofiltre/>

